The PoA of Scheduling Game with Machine Activation Costs*

Ling Lin[1,2,3], Yujie Yan[1], Xing He[1], and Zhiyi Tan[1,**]

[1] Department of Mathematics,
Zhejiang University, Hangzhou 310027, P.R. China
tanzy@zju.edu.cn
[2] School of Computer & Computing Science,
Zhejiang University City College, Hangzhou 310015, P.R. China
[3] Department of Fundamental Education, Ningbo Institute of Technology,
Zhejiang University, Ningbo 315100, P.R. China

Abstract. In this paper, we study the scheduling game with machine activation costs. A set of jobs is to be processed on identical parallel machines. The number of machines available is unlimited, and an activation cost is needed whenever a machine is activated in order to process jobs. Each job chooses a machine on which it wants to be processed. The cost of a job is the sum of the load of the machine it chooses and its shared activated cost. The social cost is the total cost of all jobs. Representing PoA as a function of the number of jobs, we get the tight bound of PoA. Representing PoA as a function of the smallest processing time of jobs, improved lower and upper bound are also given.

1 Introduction

In this paper, we study the scheduling game with machine activation costs. There is a set $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ of jobs to be processed on identical parallel machines. The processing time of J_j is p_j, $j = 1, \cdots, n$. The number of machines available is unlimited, and an activation cost B is needed whenever a machine is activated in order to process jobs. Each job chooses a machine on which it wants to be processed. The choices of all jobs determine a schedule. The *load* of a machine M_i in a schedule is the sum of the processing time of all jobs selecting M_i. The activation cost of an activated machine is shared by the jobs selecting M_i, and the amount of each job shares is proportional to its processing time. The cost of a job in the schedule is the sum of the load of the machine it chooses and its shared activated cost. A schedule is a *Nash Equilibrium* (NE) if no job can reduce its cost by neither moving to a different machine, nor activating a new machine. The game model was first proposed by [7], and it was proved that the NE always exists for any job set $\mathcal{J}$.

* Supported by the National Natural Science Foundation of China (10971191, 11271324), Zhejiang Provincial Natural Science Foundation of China (LR12A01001) and Fundamental Research Funds for the Central Universities.
** Corresponding author.

J. Chen, J.E. Hopcroft, and J. Wang (Eds.): FAW 2014, LNCS 8497, pp. 182–193, 2014.

Though the behavior of each job is influenced by individual costs, the performance of the whole system is measured by certain social cost. It is well known that in most situation NE are not optimal from this perspective due to lack of coordination. The inefficiency of NE can be measured by the *Price of Anarchy* (PoA for short) [9]. The PoA of an instance is defined as the ratio between the maximal social cost of a NE and the optimal social cost. The PoA of the game is the supremum value of the PoA of all instances.

The most favorite utilitarian social cost is the total cost of all jobs. Unfortunately, it is easily to show that the PoA of above game is infinity [2], which makes no sense. However, since PoA of the game is a kind of worst-case measure, it does not imply that the NE behaviors poorly for each job set. A common method to reveal the complete characteristic of NE in such situation is as follows: select a parameter and represent the PoA as a function of it. In [2], Chen and Gurel regard the PoA as a function of $\rho = \frac{B}{\min_{1\le j\le n} p_j}$, and prove that the PoA is at least $\frac{1}{4}(\sqrt{\rho}+2)$ and at most $\frac{1}{2}(\rho+1)$. However, the bounds are not tight.

Scheduling games with machine activation costs with different social costs were also studied in the literature. For the egalitarian social cost of minimizing the maximum cost among all jobs. Feldman and Tamir [7] proved that the PoA is $\frac{\tau+1}{2\sqrt{\tau}}$ when $\tau > 1$ and 1 when $0 < \tau \le 1$, where $\tau = \frac{B}{\max_{1\le j\le n} p_j}$. Fruitful results on scheduling games without machine activation costs can be found in [9],[6], [1], [3], [8], [4], [5].

In this paper, we revisit the scheduling game with machine activation cost with social cost of minimizing the total cost of jobs. Representing the PoA as a function of n, the number of jobs. We show that the PoA is $\frac{n+1}{3}$, and the bound is tight. We also improve the lower and upper bounds on the PoA with respect to ρ. The PoA is at most $\max\{1, \frac{\rho+1}{3}\}$, and at least $\frac{\rho+1}{2\sqrt{\rho}}$.

The paper is organized as follows. In Section 2, we give some preliminary results. In Sections 3 and 4, we present the PoA as a function of the number of jobs and the smallest processing time of the jobs, respectively.

2 Preliminaries

Let $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ be a job set. W.l.o.g., we assume $n \ge 2$, $p_1 \ge p_2 \ge \cdots \ge p_n$. By scaling the processing times we can assume that $B = 1$. Denote $P = \sum_{j=1}^{n} p_j$. Write $\rho = \frac{1}{p_n}$ and $\tau = \frac{1}{p_1}$ for simplicity. Given a schedule σ^A, the number of machines activated in σ^A is denoted m^A. Denote by $\mathcal{J}_i^A$ the set of jobs processing on M_i, $i = 1, \cdots, m^A$. The number of jobs and the total processing time of jobs of $\mathcal{J}_i^A$ are denoted n_i^A and L_i^A, respectively. Let $n_{min}^A = \min_{1\le i\le m^A} n_i^A$ and $n_{max}^A = \max_{1\le i\le m^A} n_i^A$. For any $J_j \in \mathcal{J}$, the cost of J_j in σ^A is denoted C_j^A, and the total cost of jobs of $\mathcal{J}$ in σ^A is denoted $C^A(\mathcal{J})$. Let σ^* be the optimal schedule with minimal social cost, and σ^{NE} be the worst NE, i.e., a NE with maximal social cost. W.l.o.g., we assume $J_1 \in \mathcal{J}_1^{NE}$ and $J_1 \in \mathcal{J}_1^*$. A job is called *separate* in σ^A if it is processed separately on the machine it selects.

The following three lemmas are given in [2], and are relevant to our study.

Lemma 1. *[2] (i) Any job of processing time no less than* 1 *must be a separate job in* σ^{NE}*.*

(ii) If each job selecting machine M_i *has processing time no more than* 1*, then* $L_i^{NE} \leq 1$*.*

Lemma 2. *[2] (i) Any job of processing time no less than* 1 *must be a separate job in* σ^**.*

(ii) If $n_i^* \geq 2$*, then* $L_i^* \leq 1$*.*

Lemma 3. *[2] (i)* $C^{NE} \leq m + P \leq n + P$*,*

(ii) For any job set $\mathcal{J}$*,* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{n+P}{2P\sqrt{\tau}}$*.*

Lemma 4. *If there exist* M_i *in* σ^{NE} *and* M_k *in* σ^**, such that* $\mathcal{J}_i^{NE} = \mathcal{J}_k^*$*, then* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{C^{NE}(\mathcal{J} \backslash \mathcal{J}_i^{NE})}{C^*(\mathcal{J} \backslash \mathcal{J}_i^{NE})}$*.*

Proof. Denote by n_0 and P_0 the number and the total processing time of jobs of $\mathcal{J}_i^{NE}$, respectively. Let σ' be the schedule resulting from σ^{NE} by deleting $\mathcal{J}_i^{NE}$ and M_i. Since the set of jobs selecting any machine other than M_i is not change, σ' is also a NE, and the cost of any job of $\mathcal{J} \backslash \mathcal{J}_i^{NE}$ in σ' remains the same as that in σ^{NE}. Hence,

$$\begin{aligned} C^{NE}(\mathcal{J}) &= \sum_{J_j \in \mathcal{J}_i^{NE}} C_j + \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_i^{NE}} C_j = 1 + n_0 P_0 + \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_i^{NE}} C_j' \\ &\leq 1 + n_0 P_0 + C^{NE}(\mathcal{J} \backslash \mathcal{J}_i^{NE}), \end{aligned}$$

where C_j' is the cost of J_j in σ'. On the other hand, construct a schedule $\sigma^{*'}$ from σ^* by deleting $\mathcal{J}_k^*$ and M_k. Clearly, $\sigma^{*'}$ is a feasible schedule of $\mathcal{J} \backslash \mathcal{J}_k^*$, and the cost of any job of $\mathcal{J} \backslash \mathcal{J}_k^*$ in $\sigma^{*'}$ remains the same as that in σ^*. Hence

$$\begin{aligned} C^*(\mathcal{J}) &= \sum_{J_j \in \mathcal{J}_k^*} C_j^* + \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_k^*} C_j^* = 1 + n_0 P_0 + \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_k^*} C_j^{*'} \\ &\geq 1 + n_0 P_0 + C^*(\mathcal{J} \backslash \mathcal{J}_k^*), \end{aligned}$$

where $C_j^{*'}$ is the cost of J_j in $\sigma^{*'}$. Recall that $\mathcal{J}_i^{NE} = \mathcal{J}_k^*$. Thus,

$$\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{1 + n_0 P_0 + C^{NE}(\mathcal{J} \backslash \mathcal{J}_i^{NE})}{1 + n_0 P_0 + C^*(\mathcal{J} \backslash \mathcal{J}_i^{NE})} \leq \frac{C^{NE}(\mathcal{J} \backslash \mathcal{J}_i^{NE})}{C^*(\mathcal{J} \backslash \mathcal{J}_i^{NE})}.$$

□

The main results of this paper are the following two theorems.

Theorem 1. *For any job set* $\mathcal{J}$*,* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{n+1}{3}$*, and the bound is tight.*

Theorem 2. *For any job set* $\mathcal{J}$*,* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \max\{1, \frac{\rho+1}{3}\}$*.*

Both theorems will be proved by contradiction. Suppose that there exist counterexamples. Let $\mathcal{J}$ be a minimal counterexample with the smallest number of jobs. By Lemmas 1(i), 2(i) and 4, all jobs of $\mathcal{J}$ must have processing time less than 1. Otherwise, delete any job of processing time no less than 1 results a counterexample with smaller number of jobs. Consequently, $L_i^{NE} \leq 1, i = 1, \cdots, m$ by Lemma 1(ii). On the other hand, no matter whether or not there exists a machine in σ^* which processes exactly one job, we have $L_i^* \leq 1, i = 1, \cdots, m^*$ by Lemma 2(ii). Thus

$$P \leq m^*. \tag{1}$$

Recall that $J_1 \in \mathcal{J}_1^{NE}$ and $J_1 \in \mathcal{J}_1^*$, the following lemma compares the social cost of the NE and optimal schedule of $\mathcal{J}$ and $\mathcal{J} \backslash \mathcal{J}_1^{NE}$.

Lemma 5. *(i)* $C^{NE}(\mathcal{J}) \leq 1 + n_1^{NE} L_1^{NE} + C^{NE}(\mathcal{J} \backslash \mathcal{J}_1^{NE})$.
(ii) $C^*(\mathcal{J}) \geq 2L_1^{NE} + C^*(\mathcal{J} \backslash \mathcal{J}_1^{NE})$.
(iii) If $\mathcal{J}_1^* = \{J_1\}$, *then* $C^*(\mathcal{J}) \geq 1 - p_1 + 2L_1^{NE} + C^*(\mathcal{J} \backslash \mathcal{J}_1^{NE})$.

Proof. (i) is obvious. Construct a feasible schedule $\sigma^{*'}$ of $\mathcal{J} \backslash \mathcal{J}_1^{NE}$ from σ^* by deleting all jobs of $\mathcal{J}_1^{NE}$, and all machines which only processing jobs of $\mathcal{J}_1^{NE}$. Obviously, $C_j^{*'} \leq C_j^*$ for any $J_j \in \mathcal{J} \backslash \mathcal{J}_1^{NE}$, where $C_j^{*'}$ is the cost of J_j in $\sigma^{*'}$. Thus

$$\begin{aligned} C^*(\mathcal{J}) &= \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_1^{NE}} C_j^* + \sum_{J_j \in \mathcal{J}_1^{NE}} C_j^* \geq \sum_{J_j \in \mathcal{J} \backslash \mathcal{J}_1^{NE}} C_j^{*'} + \sum_{J_j \in \mathcal{J}_1^{NE}} C_j^* \\ &\geq C^*(\mathcal{J} \backslash \mathcal{J}_1^{NE}) + \sum_{J_j \in \mathcal{J}_1^{NE}} C_j^*. \end{aligned}$$

For any $J_j \in \mathcal{J}_1^{NE}$, if J_j is a separate job of σ^*, then $C_j^* = 1 + p_j \geq 2p_j$. Otherwise, J_j is processed on the machine together with at least one other job, we also have $C_j^* \geq 2p_j$. Thus (ii) and (iii) are proved. □

3 PoA with Respect to the Number of Jobs

In this section, we will show the tight PoA as a function of n. We first give some more lemmas revealing properties of NE schedule.

Lemma 6. *If there exist machine* M_i *and* M_k *such that* $L_i^{NE} < L_k^{NE} + p_j$, *where* J_j *is the job of* $\mathcal{J}_i^{NE}$ *with the largest processing time, then* $n_i^{NE} L_k^{NE} + L_i^{NE} \geq 1$.

Proof. Note that $p_j \geq \frac{L_i^{NE}}{n_i^{NE}}$. If J_j moves to M_k, its new cost would be $C_j'' = L_k^{NE} + p_j + \frac{p_j}{L_k^{NE} + p_j}$. If $n_i^{NE} L_k^{NE} + L_i^{NE} < 1$, then

$$\begin{aligned} C_j'' - C_j^{NE} &= \left(L_k^{NE} + p_j + \frac{p_j}{L_k^{NE} + p_j} \right) - \left(L_i^{NE} + \frac{p_j}{L_i^{NE}} \right) \\ &= (L_k^{NE} + p_j - L_i^{NE}) \left(1 - \frac{p_j}{(L_k^{NE} + p_j) L_i^{NE}} \right) \end{aligned}$$

$$\leq \left(L_k^{NE} + p_j - L_i^{NE}\right)\left(1 - \frac{\frac{L_i^{NE}}{n_i^{NE}}}{(L_k^{NE} + \frac{L_i^{NE}}{n_i^{NE}})L_i^{NE}}\right)$$

$$= \left(L_k^{NE} + p_j - L_i^{NE}\right)\left(1 - \frac{1}{n_i^{NE}L_k^{NE} + L_i^{NE}}\right) < 0.$$

This contradicts that σ^{NE} is a NE. Therefore, $n_i^{NE}L_k^{NE} + L_i^{NE} \geq 1$. □

Lemma 7. *If J_j is a separate job in σ^{NE} selecting M_i, then*
(i) $L_k^{NE} \geq 1 - p_j$ for any $1 \leq k \leq m^{NE}$ and $k \neq i$.
(ii) For any job J_l with $p_l > p_j$, J_l is also a separate job in σ^{NE}.

Proof. (i) Note that $L_i^{NE} = p_j < L_k^{NE} + p_j$. By Lemma 6, $L_k^{NE} \geq \frac{1-L_i^{NE}}{n_i^{NE}} = 1 - p_j$.

(ii) Assume that J_l selects M_k together with at least one another job, say J_t. By (i), $L_k^{NE} \geq 1 - p_j$. Hence,

$$L_i^{NE}L_k^{NE} + (L_k^{NE} - 1)p_t \geq L_i^{NE}L_k^{NE} - p_jp_t = p_j(L_k^{NE} - p_t) \geq p_jp_l > 0. \quad (2)$$

If J_t moves to M_i, its new cost would be $C_t'' = L_i^{NE} + p_t + \frac{p_t}{L_i^{NE}+p_t}$. Thus by (2) and $L_i^{NE} + p_t = p_j + p_t < p_l + p_t \leq L_k^{NE}$, we have

$$C_t'' - C_t^{NE} = \left(L_i^{NE} + p_t + \frac{p_t}{L_i^{NE} + p_t}\right) - \left(L_k^{NE} + \frac{p_t}{L_k^{NE}}\right)$$

$$= (L_i^{NE} + p_t - L_k^{NE})\left(1 - \frac{p_t}{(L_i^{NE} + p_t)L_k^{NE}}\right)$$

$$= (L_i^{NE} + p_t - L_k^{NE})\frac{L_i^{NE}L_k^{NE} + (L_k^{NE} - 1)p_t}{(L_i^{NE} + p_t)L_k^{NE}} < 0.$$

It contradicts that σ^{NE} is a NE. Hence, J_l is also a separate job. □

Lemma 8. *If there exist M_i and M_k, such that $L_i^{NE} + L_k^{NE} < 1$, then $n_i^{NE} \geq 2$ and $n_k^{NE} \geq 2$.*

Proof. Assume $n_i^{NE} = 1$ and let the unique job selecting M_i be J_j. By Lemma 7(i), $L_k^{NE} \geq 1 - p_j = 1 - L_i^{NE}$, a contradiction. Hence, $n_i^{NE} \geq 2$. Similarly, we also have $n_k^{NE} \geq 2$. □

Lemma 9. *(i) If $P < 1$, then $n_i^{NE} \geq 2$ for any $1 \leq i \leq m^{NE}$ and $m^{NE} \leq \frac{n}{2}$.*
(ii) If $P < \frac{1}{n}$, then $m^{NE} = 1$.

Proof. (i) We only need to consider the case of $m^{NE} \geq 2$ due to $n \geq 2$. Since $P < 1$, the sum of the loads of any two machines are less than 1. By Lemma 8, $n_i^{NE} \geq 2$ for any $1 \leq i \leq m^{NE}$. It follows $n \geq 2m^{NE}$.

(ii) Assume that $m^{NE} \geq 2$. Let M_i and M_k be any two machines and $L_i^{NE} \leq L_k^{NE}$. By Lemma 6,

$$1 \leq n_i^{NE}L_k^{NE} + L_i^{NE} < n_i^{NE}P + P = (n_i^{NE} + 1)P < nP,$$

a contradiction. □

Lemma 10. *If $P \le 1$, then $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{n+1}{3}$.*

Proof. We distinguish several cases according to the value of m^*. If $m^* \ge 3$, then $C^*(\mathcal{J}) > m^* \ge 3$. Applying Lemma 3(i), $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} < \frac{n+P}{3} \le \frac{n+1}{3}$.

Now we turn to the case of $m^* = 2$. Clearly, $C^{NE}(\mathcal{J}) \le m^{NE} + n^{NE}_{max}P$ and $C^*(\mathcal{J}) \ge 2 + n^*_{min}P$. Define

$$\Delta_1 = (n+1)(2 + n^*_{min}P) - 3(m^{NE} + n^{NE}_{max}P).$$

To prove that $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{n+1}{3}$, it is sufficient to show $\Delta_1 \ge 0$. If $m^{NE} = 1$, then

$$\Delta_1 \ge (n+1)(2+P) - 3(1+nP) = (2n-1)(1-P) \ge 0. \tag{3}$$

Otherwise, $m^{NE} \ge 2$ and $n \ge n^{NE}_{max} + 2(m^{NE} - 1)$ by Lemma 9(i). Hence,

$$\begin{aligned}\Delta_1 &\ge (n^{NE}_{max} + 2m^{NE} - 1)(2 + n^*_{min}P) - 3(m^{NE} + n^{NE}_{max}P)\\ &= n^{NE}_{max}(2 + (n^*_{min} - 3)P) + (m^{NE} - 2) + (2m^{NE} - 1)n^*_{min}P\\ &= n^{NE}_{max}(3 - n^*_{min})(1-P) + n^{NE}_{max}(n^*_{min} - 1)\\ &\quad + (m^{NE} - 2) + (2m^{NE} - 1)n^*_{min}P.\end{aligned}$$

The last two equalities of above formula indicate that $\Delta_1 \ge 0$ no matter whether $n^*_{min} \ge 3$ or not. The proof of this case is thus completed.

For the remaining case of $m^* = 1$, $C^*(\mathcal{J}) = 1 + nP$. If $P < \frac{1}{n}$, $m^{NE} = 1$ by Lemma 9(ii). Thus $C^{NE}(\mathcal{J}) = C^*(\mathcal{J})$. Otherwise, $nP \ge 1$, and we have $n \ge 2m^{NE}$ by Lemma 9(i). Therefore,

$$\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{m^{NE} + nP}{1 + nP} \le \frac{m^{NE} + 1}{2} \le \frac{n+1}{3}.$$

□

By Lemma 10, in order to prove the Theorem 1, only the situation of $P > 1$ would be considered. If $n = n^{NE}_1 + 1$, then $n^{NE}_1 = 1$ by Lemma 7(ii). Thus $n = 2$ and $P = p_1 + p_2 > 1$. By Lemmas 1(ii) and 2(ii), $m^{NE} = m^* = 2$. Thus $C^{NE}(\mathcal{J}) = C^*(\mathcal{J})$. Hence, we have

$$n \ge n^{NE}_1 + 2. \tag{4}$$

Before proving Theorem 1, we give a technique lemma about the optimal schedule of the partial job set $\mathcal{J} \backslash \mathcal{J}^{NE}_1$.

Lemma 11. *If $P > 1$ and $n \ge n^{NE}_1 + 2$, then $C^*(\mathcal{J} \backslash \mathcal{J}^{NE}_1) \ge 3 - 2L^{NE}_1 \ge 1$.*

Proof. Let the number of machines activated in the optimal schedule of $\mathcal{J} \backslash \mathcal{J}^{NE}_1$ be m'. If $m' \ge 3$, then $C^*(\mathcal{J} \backslash \mathcal{J}^{NE}_1) > m' > 3 - 2L^{NE}_1$. If $m' = 2$, then $C^*(\mathcal{J} \backslash \mathcal{J}^{NE}_1) \ge 2 + (P - L^{NE}_1) \ge 3 - 2L^{NE}_1$. If $m' = 1$, then

$$C^*(\mathcal{J} \backslash \mathcal{J}^{NE}_1) \ge 1 + (n - n^{NE}_1)(P - L^{NE}_1) \ge 1 + 2(P - L^{NE}_1) > 3 - 2L^{NE}_1.$$

Proof of Theorem 1. Since $\mathcal{J}$ is the minimal counterexample, we have $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} > \frac{n+1}{3}$ and $\frac{C^{NE}(\mathcal{J}\backslash\mathcal{J}_1^{NE})}{C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})} \leq \frac{n-n_1^{NE}+1}{3}$. Define

$$\Delta_2 = n_1^{NE} C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE}) + (2n+2-3n_1^{NE})L_1^{NE} - 3. \tag{5}$$

Then by Lemma 5(i) and (ii),

$$\begin{aligned}\Delta_2 &= (n+1)(2L_1^{NE} + C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})) \\ &\quad -3\left(1 + n_1^{NE}L_1^{NE} + \frac{n-n_1^{NE}+1}{3}C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})\right) \\ &\leq (n+1)(2L_1^{NE} + C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})) - 3(1 + n_1^{NE}L_1^{NE} + C^{NE}(\mathcal{J}\backslash\mathcal{J}_1^{NE})) \\ &\leq (n+1)C^*(\mathcal{J}) - 3C^{NE}(\mathcal{J}) < 0.\end{aligned}$$

However, we will show below that $\Delta_2 \geq 0$, which leads contradiction.

Substituting (4) into (5), we have

$$\begin{aligned}\Delta_2 &= n_1^{NE}C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE}) + (2n_1^{NE} + 4 + 2 - 3n_1^{NE})L_1^{NE} - 3 \\ &= n_1^{NE}(C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE}) - L_1^{NE}) + 6L_1^{NE} - 3.\end{aligned}$$

If $n_1^{NE} \geq 3$, then

$$\begin{aligned}\Delta_2 &\geq 3(C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE}) - L_1^{NE}) + 6L_1^{NE} - 3 \\ &= 3C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE}) + 3L_1^{NE} - 3 \geq 0.\end{aligned}$$

Otherwise, by Lemma 11,

$$\begin{aligned}\Delta_2 &\geq n_1^{NE}(3 - 3L_1^{NE}) + 6L_1^{NE} - 3 \\ &= 3n_1^{NE} + (6 - 3n_1^{NE})L_1^{NE} - 3 \geq 3n_1^{NE} - 3 \geq 0.\end{aligned}$$

Both are contradictions.

The following instance shows that the bound is tight. Consider a job set $\mathcal{J}$ consisting of n jobs. The processing times of the jobs are $p_i = \varepsilon$, $i = 1, \cdots, n-1$ and $p_n = 1 - (n-1)\varepsilon$, where $0 < \varepsilon < \frac{1}{n-1}$. All jobs select the same machine forms a NE σ. In fact, the cost of any job of processing time ε in σ is $1+\varepsilon$, which equals to the cost that it activates a new machine. The cost of the job of processing time $1-(n-1)\varepsilon$ in σ is $2-(n-1)\varepsilon$, which also equals to the cost that it activates a new machine. Hence, $C^{NE}(\mathcal{J}) \geq 1+n$. On the other hand, consider a schedule that J_n is processed on one machine, and the other jobs are processed on the other machine. Clearly,

$$C^*(\mathcal{J}) \leq 2 + 1 - (n-1)\varepsilon + (n-1)^2\varepsilon = 3 + (n-1)^2\varepsilon - (n-1)\varepsilon.$$

Consequently, we have

$$\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \geq \frac{1+n}{3+(n-1)^2\varepsilon - (n-1)\varepsilon} \to \frac{n+1}{3}(\varepsilon \to 0).$$

□

4 PoA with Respect to the Smallest Processing Time

In this section, we will show the PoA as a function of ρ. We begin with some lemmas revealing properties of the optimal schedule.

Lemma 12. *For any* $1 \le i, k \le m^*$, *if there exists a job* $J_j \in \mathcal{J}_k^*$ *such that* $L_k^* - p_j > L_i^*$, *then* $n_i^* \ge n_k^*$.

Proof. Assume for the sake of contradiction that $n_i^* < n_k^*$. Thus $n_i^* \le n_k^* - 1$. Note that the total cost of jobs of $\mathcal{J}_i^*$ and $\mathcal{J}_k^*$ in σ^* are $1 + n_i^* L_i^*$ and $1 + n_k^* L_k^*$, respectively. Construct a schedule $\sigma^{*'}$ from σ^* by moving J_j from M_k to M_i, while the assignment of the other jobs remains unchanged. The new total cost of jobs assigned to M_i and M_k in $\sigma^{*'}$ are $1+(n_i^*+1)(L_i^*+p_j)$ and $1+(n_k^*-1)(L_k^*-p_j)$, respectively. The cost of any job of $\mathcal{J} \backslash (\mathcal{J}_i^* \cup \mathcal{J}_k^*)$ in $\sigma^{*'}$ is the same as that in σ^*. Since

$$\begin{aligned} C^{*'}(\mathcal{J}) - C^*(\mathcal{J}) &= (1 + (n_i^* + 1)(L_i^* + p_j)) + (1 + (n_k^* - 1)(L_k^* - p_j)) \\ &\quad -((1 + n_i^* L_i^*) + (1 + n_k^* L_k^*)) \\ &= n_i^* p_j + L_i^* + p_j - (n_k^* - 1)p_j - L_k^* \\ &= (n_i^* - n_k^* + 1)p_j + (L_i^* + p_j - L_k^*) < 0, \end{aligned}$$

a contradiction. □

Lemma 13. *If* J_j *is a separate job in* σ^*, *then for any job* J_l *with* $p_l > p_j$, J_l *is also a separate job in* σ^*.

Proof. Let the machine which processes J_j be M_i. Assume that J_l is processed on M_k together with at least one another job, say J_t. Then $L_k^* - p_t \ge p_l > p_j = L_i^*$. By Lemma 12, $n_k^* \le n_i^* = 1$, a contradiction. Hence J_l is also a separate job. □

Recall that $\frac{1}{\rho} = p_n \le p_j \le p_1 = \frac{1}{\tau}$ for any j. Thus

$$P \ge np_n = \frac{n}{\rho}. \tag{6}$$

If $\rho < 2$, then the sum of the processing times of any two jobs of $\mathcal{J}$ is greater than 1. Thus all jobs are separate jobs both in σ^{NE} and σ^* by Lemmas 1(ii) and 2(ii). Thus $C^{NE}(\mathcal{J}) = C^*(\mathcal{J})$. Hence, we have $\rho \ge 2$ and thus $\frac{\rho+1}{3} \ge 1$. We will analysis and exclude the situation which can not appear in the minimal counterexample in the following three lemmas.

Lemma 14. *If any one of the following conditions: (i)* $\tau \ge \frac{9}{4}$; *(ii)* $P \le 1$; *(iii)* $n_{min}^* \ge 2$ *holds, then* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{\rho+1}{3}$.

Proof. (i)By Lemma 3(ii), $\tau \ge \frac{9}{4}$ and (6), $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{n+P}{2P\sqrt{\tau}} \le \frac{n+P}{3P} = \frac{\frac{n}{P}+1}{3} \le \frac{\rho+1}{3}$.

(ii)By Theorem 2, (6) and $P \le 1$, $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \le \frac{n+1}{3} \le \frac{\rho P+1}{3} \le \frac{\rho+1}{3}$.

(iii)Clearly, $C^*(\mathcal{J}) \geq m^* + n^*_{min}P$. By Lemma 3(i), (1) and $n^*_{min} \geq 2$,

$$\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{n+P}{m^*+n^*_{min}P} = \frac{\frac{n}{P}+1}{\frac{m^*}{P}+n^*_{min}} \leq \frac{\rho+1}{1+n^*_{min}} \leq \frac{\rho+1}{3}.$$

□

By Lemma 14, we assume that $\tau < \frac{9}{4}$, $P > 1$ and $n^*_{min} = 1$ in the following. Since J_1 is the job with the largest processing time, and there is at least one machine in σ^* which processes exactly one job. We have

$$\mathcal{J}^*_1 = \{J_1\} \tag{7}$$

by Lemma 13. Similarly, if $n^{NE}_{min} = 1$, then $\mathcal{J}^{NE}_1 = \{J_1\}$ according to Lemma 7(ii). Thus $\mathcal{J}^{NE}_1 = \mathcal{J}^*_1$ and $\mathcal{J}\backslash\mathcal{J}^{NE}_1$ is also a counterexample by Lemma 4, which contradicts the definition of $\mathcal{J}$. Hence, we have $n^{NE}_1 \geq n^{NE}_{min} \geq 2$, and

$$1 \geq L^{NE}_1 \geq p_1 + (n^{NE}_1 - 1)p_n = p_1 + (n^{NE}_1 - 1)\frac{1}{\rho} = \frac{1}{\tau} + \frac{n^{NE}_1 - 1}{\rho}. \tag{8}$$

Lemma 15. *If any one of the following two conditions: (i)* $2 \leq \rho < 3$*; (ii)* $3 \leq \rho < 4$ *and* $n^{NE}_1 = 3$ *holds, then* $n^*_i \leq 2$ *for any* $1 \leq i \leq m^*$.

Proof. If $2 \leq \rho < 3$, then $3p_n = \frac{3}{\rho} > 1 \geq L^*_i$. The result clearly follows. If $3 \leq \rho < 4$ and $n^{NE}_1 = 3$, then $L^*_1 = p_1 \leq 1 - \frac{2}{\rho} < \frac{2}{\rho} = 2p_n$ by (8). Assume that there exists a machine M_k, $k \geq 2$ such that $n^*_k \geq 3$, and job $J_j \in \mathcal{J}^*_k$. Then $L^*_k - p_j \geq 2p_n \geq L^*_i$, and $n^*_1 \geq n^*_k = 3$ by Lemma 12, contradicts (7). □

Lemma 16. *If any one of the following two conditions: (i)* $2 \leq \rho < 3$*; (ii)* $3 \leq \rho < 4$ *and* $n^{NE}_1 = 3$ *holds, then* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{\rho+1}{3}$.

Proof. When $2 \leq \rho < 4$, then $4p_n = \frac{4}{\rho} > 1$. Hence $2 \leq n^{NE}_{min} \leq n^{NE}_i \leq 3$ for any $1 < i \leq m^{NE}$. Consider a subset of machines $\mathcal{M}_3$ which consists of machines that exactly three jobs select in σ^{NE}. Denote by m_3 and P_3 the number of machines of $\mathcal{M}_3$ and the total processing time of jobs selecting one machine of $\mathcal{M}_3$, respectively. Then

$$n = 3m_3 + 2(m^{NE} - m_3) \geq 3m_3 \tag{9}$$

and

$$\begin{aligned} C^{NE}(\mathcal{J}) &= m_3 + 3P_3 + (m^{NE} - m_3) + 2(P - P_3) \\ &= m_3 + 3P_3 + \frac{n - 3m_3}{2} + 2(P - P_3) = \frac{n}{2} + 2P + P_3 - \frac{m_3}{2}. \end{aligned} \tag{10}$$

On the other hand, $1 \leq n^*_i \leq 2$ for any $1 \leq i \leq m^*$ by Lemma 15. Consider the subset of machines $\mathcal{M}^*_1$ which consists of machines processing exactly one job in σ^*. Denote by m^*_1 and P^*_1 the number of machines of $\mathcal{M}^*_1$ and the total processing time of jobs processed on one machine of $\mathcal{M}_1^*$, respectively. Then $n = m_1^* + 2(m^* - m_1^*)$ and

$$\begin{aligned} C^*(\mathcal{J}) &= m_1^* + P_1^* + (m^* - m_1^*) + 2(P - P_1^*) \\ &= m_1^* + P_1^* + \frac{n - m_1^*}{2} + 2(P - P_1^*) = \frac{n}{2} + 2P + \frac{m_1^*}{2} - P_1^*. \end{aligned} \quad (11)$$

Define

$$\Delta_3 = (\rho + 1)C^*(\mathcal{J}) - 3C^{NE}(\mathcal{J}).$$

In order to prove $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \leq \frac{\rho+1}{3}$, it is sufficient to prove that $\Delta_3 \geq 0$. By (10), (11),

$$\begin{aligned} \Delta_3 &= (\rho + 1)\left(\frac{n}{2} + 2P + \frac{m_1^*}{2} - P_1^*\right) - 3\left(\frac{n}{2} + 2P + P_3 - \frac{m_3}{2}\right) \\ &= (\rho - 2)\left(\frac{n}{2} + 2P\right) + (\rho + 1)\left(\frac{m_1^*}{2} - P_1^*\right) - 3\left(P_3 - \frac{m_3}{2}\right). \end{aligned} \quad (12)$$

(i) If $2 \leq \rho < 3$, then $3p_n = \frac{3}{\rho} > 1$. Thus $\mathcal{M}_3 = \emptyset$, $P_3 = 0$ and $n_1^{NE} = 2$. By (8), $p_1 \leq 1 - \frac{1}{\rho}$. Thus,

$$P_1^* \leq p_1 m_1^* \leq \frac{\rho - 1}{\rho} m_1^*, \quad (13)$$

and

$$P \geq P_1^* + (n - m_1^*)p_n = P_1^* + \frac{n - m_1^*}{\rho} \quad (14)$$

Substituting (13), (14) to (12), we have

$$\begin{aligned} \Delta_3 &= (\rho - 2)\left(\frac{n}{2} + 2P\right) + (\rho + 1)\left(\frac{m_1^*}{2} - P_1^*\right) \\ &\geq (\rho - 2)\left(\frac{n}{2} + 2P_1^* + \frac{2(n - m_1^*)}{\rho}\right) + (\rho + 1)\left(\frac{m_1^*}{2} - P_1^*\right) \\ &= (\rho - 2)\left(\frac{n}{2} + \frac{2(n - m_1^*)}{\rho}\right) + (\rho - 5)P_1^* + \frac{\rho + 1}{2} m_1^* \\ &\geq (\rho - 2)\left(\frac{n}{2} + \frac{2(n - m_1^*)}{\rho}\right) + (\rho - 5)P_1^* + \frac{\rho + 1}{2}\frac{\rho}{\rho - 1} P_1^* \\ &= (\rho - 2)\left(\frac{n}{2} + \frac{2(n - m_1^*)}{\rho}\right) + \frac{(\rho - 2)(3\rho - 5)}{2(\rho - 1)} P_1^* \geq 0. \end{aligned}$$

(ii) If $3 \leq \rho < 4$ and $n_1^{NE} = 3$, then $p_1 \leq 1 - \frac{2}{\rho} < \frac{1}{2}$ by (8). Thus $P_1^* \leq \frac{m_1^*}{2}$. Moreover, $P_3 \leq m_3$ since $L_i^{NE} \leq 1$. Together with (9), we have

$$\begin{aligned} \Delta_3 &\geq (\rho - 2)\left(\frac{n}{2} + 2P\right) - 3\left(P_3 - \frac{m_3}{2}\right) \geq (\rho - 2)\left(\frac{n}{2} + 2P\right) - \frac{3m_3}{2} \\ &\geq \frac{n}{2} - \frac{3m_3}{2} + 2P \geq 0. \end{aligned}$$

□

Proof of Theorem 2. Since $\mathcal{J}$ is the minimal counterexample, $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} > \frac{\rho+1}{3}$. Moreover, the smallest processing time of jobs of $\mathcal{J}\backslash\mathcal{J}_1^{NE}$ is no smaller than p_n. Hence, $\frac{C^{NE}(\mathcal{J}\backslash\mathcal{J}_1^{NE})}{C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})} \leq \frac{\rho+1}{3}$. Define

$$\Delta_4 = (\rho+1)(L_1^{NE} - p_1) + (\rho + 1 - 3n_1^{NE})L_1^{NE} + \rho - 2.$$

Then by Lemma 5(i), (iii),

$$\begin{aligned}\Delta_4 &= (\rho+1)(1 + 2L_1^{NE} - p_1 + C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})) \\ &\quad -3\left(1 + n_1^{NE}L_1^{NE} + \frac{\rho+1}{3}C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})\right) \\ &\leq (\rho+1)(1 + 2L_1^{NE} - p_1 + C^*(\mathcal{J}\backslash\mathcal{J}_1^{NE})) \\ &\quad -3(1 + n_1^{NE}L_1^{NE} + C^{NE}(\mathcal{J}\backslash\mathcal{J}_1^{NE})) \\ &\leq (\rho+1)C^*(\mathcal{J}) - 3C^{NE}(\mathcal{J}) < 0.\end{aligned}$$

However, we will show below that $\Delta_4 \geq 0$, which leads contradiction.

If $\rho + 1 \geq 3n_1^{NE}$, then $\Delta_4 \geq 0$. Otherwise, $\rho < 3n_1^{NE} - 1$. We distinguish several cases according to the value of n_1^{NE}. If $n_1^{NE} = 2$, then $\rho < 5$. By (8) and $\rho \geq 3$,

$$\Delta_4 \geq (\rho+1)\frac{1}{\rho} + \rho - 5 + \rho - 2 = \frac{2\rho^2 - 6\rho + 1}{\rho} \geq 0.$$

If $n_1^{NE} = 3$, then $\rho < 8$. We only need to consider the case of $\rho \geq 4$ by Lemma 16. By (8),

$$\Delta_4 \geq (\rho+1)\frac{2}{\rho} + \rho - 8 + \rho - 2 = \frac{2\rho^2 - 8\rho + 2}{\rho} \geq 0.$$

If $n_1^{NE} \geq 4$, we have $\rho \geq \frac{n_1^{NE}-1}{1-\frac{1}{\tau}} \geq \frac{9}{5}(n_1^{NE} - 1) \geq \frac{27}{5}$ by (8). Hence,

$$\begin{aligned}\Delta_4 &\geq (\rho+1)\frac{n_1^{NE} - 1}{\rho} + (\rho + 1 - 3n_1^{NE}) + \rho - 2 \\ &= \frac{1-2\rho}{\rho}n_1^{NE} + 2\rho - 1 - \frac{\rho+1}{\rho} \\ &\geq \frac{1-2\rho}{\rho}\left(\frac{5\rho}{9} + 1\right) + 2\rho - 1 - \frac{\rho+1}{\rho} = \frac{8\rho - 31}{9} \geq 0.\end{aligned}$$

□

The bound given in Theorem 2 may not be tight. Currently, we only have the following lower bound on the PoA.

Theorem 3. *For any square number ρ, there exists a job set $\mathcal{J}$, such that* $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \geq \frac{\rho+1}{2\sqrt{\rho}}$.

Proof. Consider a job set $\mathcal{J}$ consists of ρ jobs. All jobs have processing times $\frac{1}{\rho}$. All jobs select the same machine forms a NE σ. In fact, the cost of any job is $1 + \frac{1}{\rho}$, which equals to the cost that it activates a new machine. Hence, $C^{NE}(\mathcal{J}) \geq \rho + 1$. On the other hand, consider a schedule that every $\sqrt{\rho}$ jobs are processed on one machine. Clearly, $C^*(\mathcal{J}) \leq \sqrt{\rho} + \rho\frac{\sqrt{\rho}}{\rho} = 2\sqrt{\rho}$. Consequently, we have $\frac{C^{NE}(\mathcal{J})}{C^*(\mathcal{J})} \geq \frac{\rho+1}{2\sqrt{\rho}}$. □

References

1. Berenbrink, P., Goldberg, L., Goldberg, P., Martin, R.: Utilitarian resource assignment. Journal of Discrete Algorithm 4, 567–587 (2006)
2. Chen, B., Gürel, S.: Resource allocation games of utilitarian social objectives. Journal of Scheduling 15, 157–164 (2012)
3. Czumaj, A., Vöcking, B.: Tight bounds for worst-case equilibria. ACM Transactions on Algorithms 3(4) (2007)
4. Epstein, L.: Equilibria for two parallel links: the strong price of anarchy versus the price of anarchy. Acta Informatica 47, 375–389 (2010)
5. Epstein, L., van Stee, R.: The price of anarchy on uniformly related machines revisited. Information and Computation 212, 37–54 (2012)
6. Feldmann, R., Gairing, M., Lücking, T., Monien, B., Rode, M.: Nashification and the coordination ratio for a selfish routing game. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 514–526. Springer, Heidelberg (2003)
7. Feldman, M., Tamir, T.: Conflicting congestion effects in resource allocation games. Operations Research 60, 529–540 (2012)
8. Fotakis, D., Kontogiannis, S., Koutsoupias, E., Mavronicolas, M., Spirakis, P.: The structure and complexity of Nash equilibria for a selfish routing game. Theoretical Computer Science 410, 3305–3326 (2009)
9. Koutsoupias, E., Papadimitriou, C.H.: Worst-case equilibria. Computer Science Review 3, 65–69 (2009)